

Real-time Operating Systems and Systems Programming

Real Time Operating Systems

Overview

- OSE
- OSEK/VDX
- Nucleus
- VRTX
- VxWorks
- QNX
- μ C/OS

OSE by ENEA

- Operating System Embedded
- One of the most widely used RT operating systems
- 1.5 billion run-times deployed
- Swedish company
- Characterized by event messages for interprocess communication
- Wide support for processors
- Over half of 3G base stations

OSEK/VDX

- Automotive standard with many implementations (consider POSIX)
- *Offene Systeme und deren Schnittstellen für die Elektronik in Kraftfahrzeugen*
- “Open Systems and their Interfaces for the Electronics in Motor Vehicles”
- Runs without memory protection
- Open standard

OSEK/VDX Specifies

- Static tasks, stacks, mutexes
- Tasks
 - Basic: never block
 - Enhanced: can wait on objects
- Priorities, round robin for same level
- Deadlock prevention with Priority Ceiling
 - Task which holds a priority object will prevent lower priority tasks from running

Nucleus

- 2.1 billion run-times deployed (according to developer)
- Closed source
- Windowing system
- Products
 - Motorola, Samsung, LG cellular phones
 - Creative Zen soundcards
 - Many iPhone clones

VRTX

- Versatile Real-Time Executive
- Two kernels: microcontroller and scalable
- Runs Hubble Space Telescope
- Also used by Motorola

VxWorks

- Started as improvement on VTRX (1985)
- Kernel was replaced later
- Name is probably a pun on VTRX
- Development done on host such as Linux, Unix, Windows with cross-compiling to ease testing
- Honda ASIMO; Boeing 787, 747-8; Linksys WRT54G, Canon DIGIC-II & DIGIC-III; Apache Longbow attack helicopter; Spirit&Opportunity rovers

QNX

- Microkernel architecture
- cars, mobile phones
- Owned by BlackBerry
- Amount of (optional) "services"
- proc – system service for task management
- message passing passes CPU time

μ C/OS-II & -III

- Started as a two part article in Embedded Systems Programming magazine
- Free for noncommercial use
- \$10000 per end product otherwise
- Also certified for use in safety critical contexts
- The Book!

How to Choose?

- Main characteristics:
 - Worst case performance documented
 - Interrupt latency known
 - Context switch time
- Exclude from the list:
 - **Processor**
 - **Real-time performance**
 - **Price**

Last optimization

- Compatibility of compiler
- Debugging issues
- Development tools
- Memory requirements
- Add-on software (network, usb, filesystems)
- Vendor experience